

ЛАБОРАТОРНАЯ РАБОТА

Работа с системой контроля версий

Цель:

Изучить основы работы с системой контроля версий.

Работа с Git и GitHub

Основные термины:

Система контроля версий – это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определенной версии.

Git — система контроля и управления версиями файлов.

GitHub — веб-сервис для размещения репозитория и совместной разработки проектов.

Ветка (Branch) — это параллельная версия репозитория. Она включена в этот репозиторий, но не влияет на главную версию. После внесения нужных изменения, можно объединить их с главной версией.

Репозиторий Git — каталог файловой системы, в котором находятся: файлы конфигурации, файлы журналов операций, выполняемых над репозиторием, индекс расположения файлов и хранилище, содержащее сами контролируемые файлы.

Локальный репозиторий — репозиторий, расположенный на локальном компьютере разработчика в каталоге. Именно в нём происходит разработка и фиксация изменений, которые отправляются на удалённый репозиторий.

Удалённый репозиторий — репозиторий, находящийся на удалённом сервере. Это общий репозиторий, в который приходят все изменения и из которого забираются все обновления.

Fork — копия репозитория. Его также можно рассматривать как внешнюю ветку для текущего репозитория.

Клонирование (Clone) — скачивание репозитория с удалённого сервера на локальный компьютер в определённый каталог для дальнейшей работы с этим каталогом как с репозиторием.

Мастер (Master) — главная или основная ветка репозитория.

Commit — фиксация изменений или запись изменений в репозиторий.

Pull — получение последних изменений с удалённого сервера репозитория.

Push — отправка всех неотправленных коммитов на удалённый сервер репозитория.

Pull Request — запрос на слияние копии репозитория с основным репозиторием. Данный запрос может быть принят или отклонён вами, как владельцем репозитория.

Merge — слияние изменений из какой-либо ветки репозитория с любой веткой этого же репозитория. Чаще всего слияние изменений из ветки репозитория с основной веткой репозитория.

Кодревью — процесс проверки кода на соответствие определённым требованиям, задачам и внешнему виду.

Выполнение работы:

1. Создать профиль на github.com.

2. Скачать и установить Git:

Скачиваем установочный файл с сайта <https://gitforwindows.org/> и запускаем его. Настройки не меняем.

3. На рабочем столе создать две папки с именами User1 и User2.

4. Открыть командную строку:

«Пуск» → «Выполнить» → в открывшемся окне написать «cmd» → нажать «ОК» или нажать комбинацию клавиш «win + R» → «cmd» → нажать «ОК».

5. Указать имя и почту пользователя.

```
git config --global user.name "Фамилия Имя студента"
```

```
git config --global user.email "Почта студента"
```

Эти данные будут включены в каждый коммит.

6. Посмотрите текущие настройки с помощью команды

```
git config --list
```

7. Вызовите справку с помощью команды

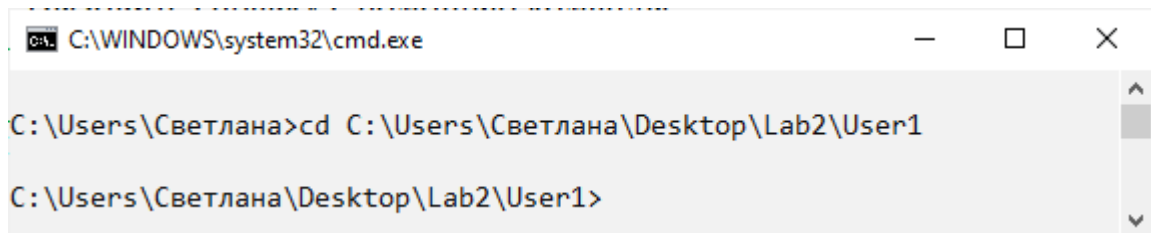
```
git help
```

и ознакомьтесь с ней.

8. В командной строке перейдите в каталог User1 с помощью команды

```
cd путь_к_каталогу
```

Пример:



9. Создать структуру репозитория с помощью команды

```
git init
```

Эта команда создаёт в текущем каталоге новый подкаталог с именем `.git`, содержащий все необходимые файлы репозитория — структуру Git репозитория.

Подкаталог `.git` является скрытым. Для того, что бы его увидеть необходимо настроить параметры папок:

«Пуск» → «Панель управления» → «Параметры проводника» (или «Параметры папок», зависит от версии Windows) → «Вид» → «Показывать скрытые файлы, папки и диски».

10. В каталоге User1 создать два текстовых файла (file1.txt и file2.txt) с произвольным текстом.

11. В командной строке посмотрите статус репозитория с помощью команды

```
git status
```

```
C:\WINDOWS\system32\cmd.exe

C:\Users\Светлана\Desktop\Lab2\User1>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.txt
        file2.txt

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\Светлана\Desktop\Lab2\User1>
```

* Untracked files – файлы, которые ещё не были добавлены в репозиторий. Изменения в таких файлах не отслеживаются.

12. Добавить содержимое текущего каталога в индекс (staging area) с помощью команды

`git add .`

Символ « . » в данной команде означает добавление всех файлов.

Команда `git add` добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита (если проще, то данная команда индексирует файлы для отслеживания изменений, т.е. делает их отслеживаемыми).

13. Используйте команду `git status`. Что изменилось?

14. Откройте файл `file2.txt`. Внесите в него изменения.

15. Используйте команду `git status`. Что изменилось?

16. Проиндексируйте измененный файл с помощью команды

`git add имя_файла`

17. Выполните первый `commit`.

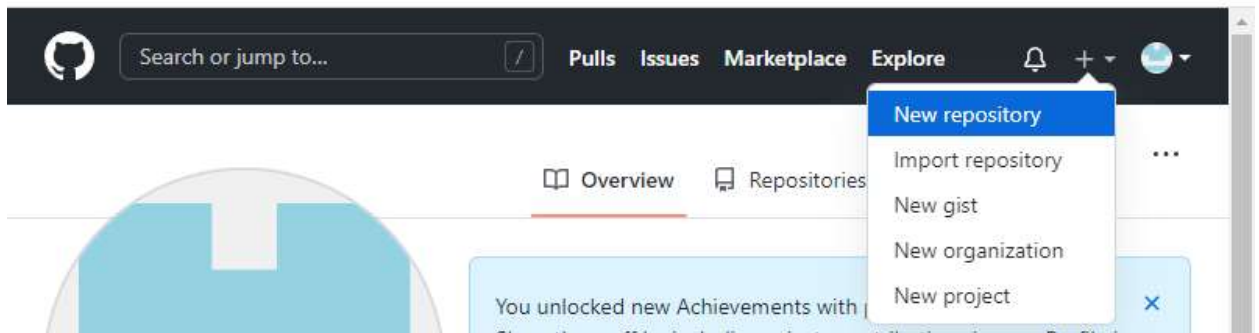
Внимание! Перед выполнением данного пункта проверьте, что все файлы в вашем каталоге проиндексированы. Изменения, которые были сделаны после последней индексации (`git add`) в `commit` не попадут.

Для создания `commit` используйте команду

```
git commit -m "first commit"
```

Параметр `-m` позволяет в кавычках оставить сообщение (комментарий) к текущему коммиту. Комментарий записывается в кавычках.

18. На сайте github.com создайте новый репозиторий:



Задайте имя репозитория (Test) и описание. Сделайте его публичным (public).

19. Подключитесь к удаленному репозиторию и загрузите файлы

Определите удалённый репозиторий, дайте ему имя origin

```
git remote add origin ссылка_на_репозиторий
```

Ссылку на репозиторий можно увидеть на сайте [github](https://github.com) после создания пустого репозитория.

Создайте ветку main

```
git branch -M main
```

Загрузите данные из каталога User1 в удаленный репозиторий

```
git push -u origin main
```

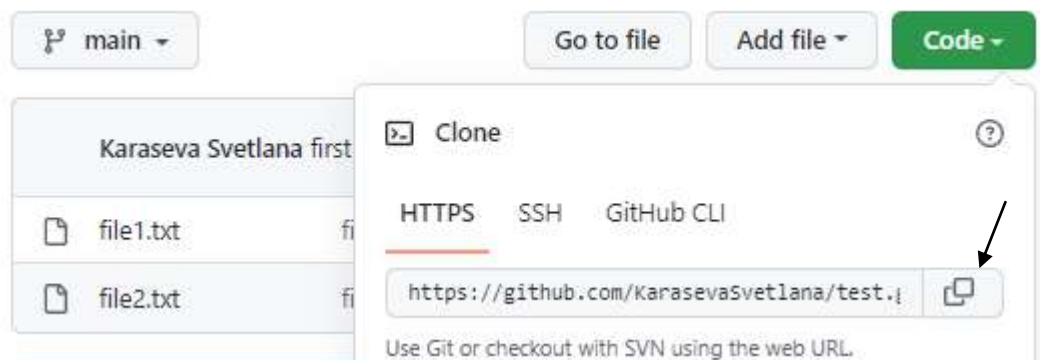
При подключении к удалённому репозиторию через командную строку необходимо будет ввести логин и пароль

20. В командной строке перейдите к каталогу User2.

Скачайте в каталог User2 файлы из созданного репозитория с помощью команды

```
git clone ссылка_на_репозиторий
```

Ссылку можно скопировать на сайте github.com, открыв нужный репозиторий:



21. В скачанную папку добавьте ещё один файл (file3.txt) и запишите в него любой текст. Из файла file1.txt удалите весь имеющийся текст и запишите новый.

22. В командной строке переместитесь в скачанный каталог Test с помощью команды

```
cd Test
```

23. Просмотрите статус каталога. Если есть неотслеживаемые файлы, проиндексируйте их.

24. Выполните commit.

Если выполнить commit без комментария, то откроется окно, изображенное на рисунке ниже:



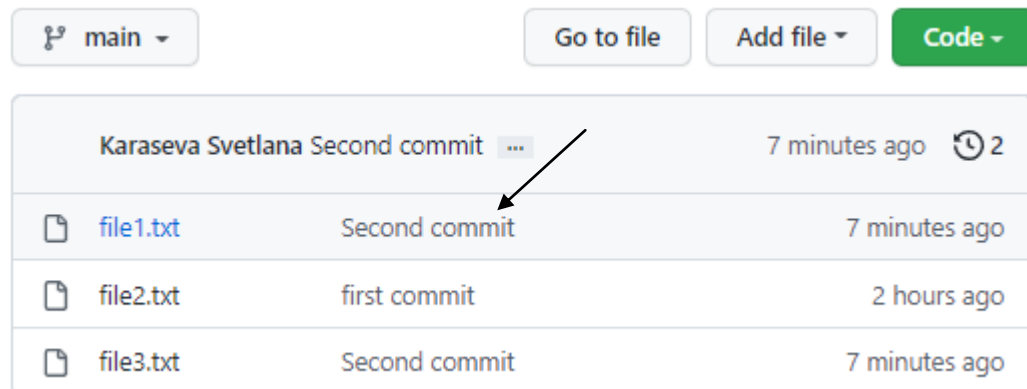
Для того чтобы закрыть его нажмите «esc», затем напишите «:wq» и нажмите «enter». Затем выполните commit с комментарием (параметром -m).

25. Загрузите файл в удаленный репозиторий с помощью команды

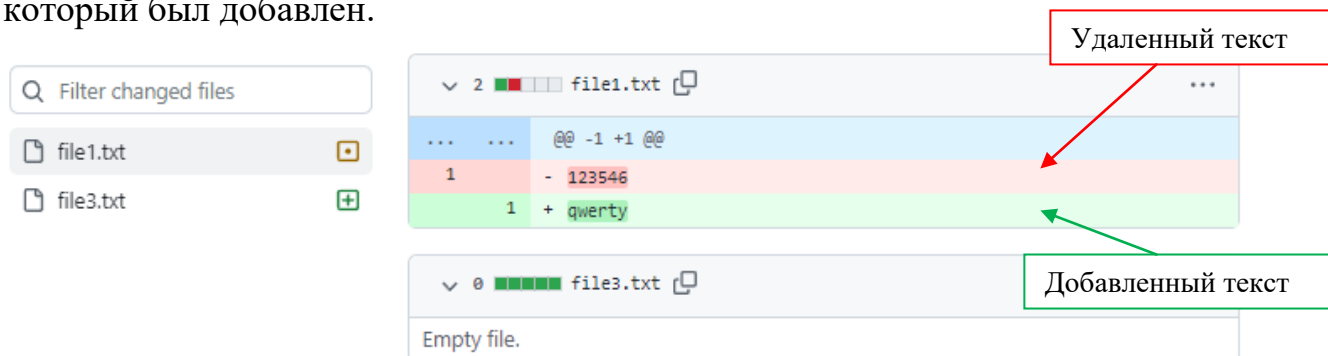
`git push`

26. Зайдите в удалённый репозиторий (при необходимости обновите страницу браузера) и убедитесь, что изменения были загружены.

При нажатии на комментарий можно увидеть, какие изменения были внесены.



Красным выделен текст, который был удален. Зеленым выделен текст, который был добавлен.



27. В командной строке перейдите в каталог User1 и скачайте обновления с помощью команды

`git pull`

Убедитесь, что обновления были загружены.

28. В каталоге User1 добавьте ещё один не пустой файл, удалите file3.txt и внесите изменения в file2.txt.

29. Загрузите изменения в удаленный репозиторий.

30. С помощью командной строки скачайте изменения в каталог User2\test.

В качестве ответа представьте преподавателю отчёт и ссылку на удаленный репозиторий.

Отчёт:

В качестве отчёта представить документ с ответами на вопросы и скриншотами (иллюстрациями), подтверждающими успешное выполнения всех заданий с личным репозиторием студента.